Report No. UIUCDCS-R-2006-2748 UILU-ENG-2006-1788

Regularized Locality Preserving Projections with Two-Dimensional Discretized Laplacian Smoothing

by

Deng Cai, Xiaofei He, and Jiawei Han

July 2006

Regularized Locality Preserving Projections with Two-Dimensional Discretized Laplacian Smoothing

Deng Cai, Xiaofei He, and Jiawei Han, *Senior Member, IEEE*

***Abstract*—A novel approach to linear dimensionality reduction is introduced that is based on Locality Preserving Projections (LPP) with a discretized Laplacian smoothing term. The choice of penalty allows us to incorporate prior information that some features may be correlated. For example, an $n_1 \times n_2$ image represented in the plane is intrinsically a matrix. The pixels spatially close to each other may be correlated. Even though we have $n_1 \times n_2$ pixels per image, this spatial correlation suggests the real number of freedom is far less. However, most of the previous methods consider an image as a vector in $\mathbb{R}^{n_1 \times n_2}$. They do not take advantage of the spatial correlation in the image, and the pixels are considered as independent. In this paper, we introduce a Regularized LPP model using a Laplacian penalty to constrain the coefficients to be spatially smooth. By preserving the local geometrical structure of the image space, we can obtain a linear subspace which is optimal for image representation in the sense of local isometry. Recognition, clustering and retrieval can be then performed in the image subspace. Experimental results on face representation and recognition demonstrate the effectiveness of our method.**

I. Introduction

Recently there are considerable interest in geometrically motivated approaches to visual analysis. The visual data like image and video is generally of very high dimensionality, ranging from several thousands to several hundreds of thousands. For example, a typical image of face is of size 32×32, resulting in a 1024-dimensional vector. However, the intrinsic degrees of freedom is far less. Various researchers (see [3], [5], [35], [37], [49]) have considered the case when the data lives on or close to a submanifold of the ambient space. One hopes then to estimate geometrical and topological properties of the submanifold from random points ("scattered data") lying on this unknown submanifold.

Previous works have demonstrated that the face recognition performance can be improved significantly in lower dimensional linear subspaces [2], [21], [30], [32], [38], [41]. Two of the most popular appearance-based face recognition methods include *Eigenface* [38] and *Fisherface* [2]. Eigenface is based on Principal Component Analysis (PCA) [13]. PCA projects the face images along the directions of maximal variances. It also aims to preserve the Euclidean distances between face images. For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produces a compact representation. Fisherface is based on Linear Discriminant Analysis (LDA) [13]. Unlike PCA which is unsupervised, LDA is supervised. When the class information is available, LDA can be used to find a linear subspace which is optimal for discrimination. Some extensions and variants of PCA and LDA have also been proposed, such as Penalized Discriminant Analysis [18], Kernel PCA [36], Kernel LDA [1], [44], etc.

Recently, the Locality Preserving Projection (LPP) algorithm is proposed to discover the local geometrical structure of the data space [20]. LPP is derived by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the data manifold. The Laplace Beltrami operator takes the second order derivatives of the functions on the manifolds. It measures the smoothness of the functions. Therefore, LPP can discover the nonlinear manifold structure to some extent. LPP has demonstrated its effectiveness in face recognition. The basis functions obtained by LPP is generally referred to as *Laplacianfaces* [21].

All the above methods consider a face image as a high dimensional vector. They do not take advantage of the spatial correlation in the image, and the pixels are considered as independent pieces of information. However, a $n_1 \times n_2$ face image represented in the plane is intrinsically a matrix, or 2-order tensor. Even though we have $n_1 \times n_2$ pixels per iamge, this spatial correlation suggests the real number of freedom is far less. Recently there has been a lot of interest in tensor based approaches to data analysis in high dimensional spaces. Vasilescu and Terzopoulos have proposed a novel face representation algroithm called Tensorface [40]. Tensorface represents the set of face images by a higher-order tensor and extends Singular Value Decomposition (SVD) to higher-order tensor data. Some other researchers have also shown how to extend PCA, LDA, and LPP to higher order tensor data [7], [19], [42], [43], [46], [47]. Some experimental results have showed the superiority of these tensor approaches over their corresponding vector approaches. However, our analysis later will show that these tensor approaches only consider the relationship between pixels in the same row (column) and fail to fully explorer the spatial information of images. The embedding functions of tensor approaches will still be spatially rough.

In this paper, we introduce a Regularized LPP (RLPP) model using a Laplacian penalty to constrain the coefficients to be spatially smooth. Instead of considering the basis function as a $n_1 \times n_2$-dimensional vector, we consider it as a matrix, or a discrete function defined on a $n_1 \times n_2$ lattice. Thus, the discretized Laplacian can be applied to the basis functions to measure their smoothness along horizontal and vertical directions. The discretized Laplacian operator is a finite difference

D. Cai and J. Han are with the Department of Computer Science, University of Illinois at Urbana Champaign, 201 N. Goodwin Ave., Urbana, IL 61801. Email: {dengcai2, hanj}@cs.uiuc.edu.

X. He is with Yahoo Research Labs, 3333 Empire Ave., Burbank, CA 91506. Email: hex@yahoo-inc.com.

approximation to the second derivative operator, summed over all directions. The choice of Laplacian penalty allows us to incorporate the prior information that neighboring pixels are correlated.

Once we obtain compact representations of the images, classification and clustering can be performed in the lower dimensional subspace.

The points below highlight several aspects of the paper:

1) When the number-of-dimensions to sample-size ratio is too high, it is difficult for LPP to discover the intrinsic geometrical structure. Since the image data generally has a large number of dimensions (pixels), natural methods of regularization emerge.
2) Even if the sample size were sufficient to estimate the intrinsic geometrical structure, coefficients of spatially smooth features (pixels) tend to be spatially rough. Since we hope to interpret these coefficients, we would prefer smoother versions, especially if they do not compromise the fit.
3) The primary focus of this paper is on face images. However, our method can be naturally extended to higher order tensors, such as videos which are intrinsically the third order tensors. Our results may also be of interest to researchers in computer graphics who have considered the question of modeling the Bidirectional Texture Function (BTF) whose observational data is of six dimensions (i.e. sixth order tensor), two variables for surface location, two variables for view direction and two variables for illumination direction [31]. Researchers in computer vision, pattern recognition, molecular biology, information retrieval, and other areas where large amount of higher order tensor (rather than vector) based data are available may find some use of the algorithm and analysis of this paper.

The remainder of the paper is organized as follows. In Section 2, we provide a brief review of PCA, LDA, LPP and their tensor extensions. Section 3 introduces our proposed Regularized LPP with two dimensional discretized Laplacian smoothing algorithm. Section 4 provides the theoretical analysis of our algorithm. In Section 5, we describe the Smooth Laplacianfaces approach for face representation. The extensive experimental results are presented in Section 6. Finally, we provide some concluding remarks and suggestions for future work in Section 7.

II. PCA, LDA, LPP and Their Tensor Extensions

Suppose we have m $n_1 \times n_2$ face images. Let $\{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^n$ ($n = n_1 \times n_2$) denote their vector representations and $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$.

A. PCA

PCA is a canonical linear dimensionality reduction algorithm. The basic idea of PCA is to project the data along the directions of maximal variances so that the reconstruction error can be minimized. Let $\mathbf{a}$ be the projection vector and $y_i = \mathbf{a}^T\mathbf{x}_i$. Let $\boldsymbol{\mu} = \frac{1}{m}\sum \mathbf{x}_i$ and $\overline{y} = \frac{1}{m}\sum y_i$. The objective function of PCA is as follows:

$$\begin{aligned}\mathbf{a}_{opt} &= \arg\max_{\mathbf{a}} \sum_{i=1}^m (y_i - \overline{y})^2 \\ &= \arg\max_{\mathbf{a}} \sum_{i=1}^m \mathbf{a}^T (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{a} \\ &= \arg\max_{\mathbf{a}} \mathbf{a}^T C \mathbf{a}\end{aligned}$$

where $C = \frac{1}{m}\sum_{i=1}^m (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T$ is the data covariance matrix. The basis functions of PCA are the eigenvectors of the data covariance matrix associated with the largest eigenvalues.

B. LDA

Unlike PCA which is unsupervised, LDA is supervised. Suppose we have c classes and the i-th class have m_i samples, $m_1 + \cdots + m_c = m$. Let $\boldsymbol{\mu}_i$ be the sample mean vector of the i-th class. LDA aims to maximize the ratio of between-class variance to the within-class variance thereby guaranteeing maximal separability. The objective function of LDA is as follows:

$$\max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}} \tag{1}$$

where S_b is the *between-class scatter matrix* and S_w is the *within-class scatter matrix*. They are defined as follows:

$$S_b = \sum_{i=1}^c m_i (\boldsymbol{\mu}^i - \boldsymbol{\mu})(\boldsymbol{\mu}^i - \boldsymbol{\mu})^T$$

$$S_w = \sum_{i=1}^c \left(\sum_{j=1}^{m_i} (\mathbf{x}_j^i - \boldsymbol{\mu}^i)(\mathbf{x}_j^i - \boldsymbol{\mu}^i)^T \right)$$

where $\mathbf{x}_j^i$ is the j-th sample in the i-th class. Thus, the basis functions of LDA that maximize the objective function is given by the maximum eigenvalue solution to the generalized eigenvalue problem:

$$S_b\mathbf{a} = \lambda S_w \mathbf{a} \tag{2}$$

We can define the *total scatter matrix* as:

$$S_t = \sum_{i=1}^m (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T = mC$$

It is easy to verify that $S_t = S_b + S_w$ [14], thus:

$$\begin{aligned} & S_b\mathbf{a} = \lambda S_w\mathbf{a} \\ \Rightarrow\ & (S_t - S_w)\mathbf{a} = \lambda S_w \mathbf{a} \\ \Rightarrow\ & S_w\mathbf{a} = \frac{1}{1+\lambda} S_t \mathbf{a} \\ \Rightarrow\ & S_b\mathbf{a} = \lambda (S_t - S_b)\mathbf{a} \\ \Rightarrow\ & S_b\mathbf{a} = \frac{\lambda}{1+\lambda} S_t\mathbf{a} \end{aligned}$$

Therefore, LDA can also be obtained by solving the following *minimum* eigenvalue problem:

$$S_w\mathbf{a} = \lambda S_t \mathbf{a} \tag{3}$$

or the following *maximum* eigenvalue problem:

$$S_b\mathbf{a} = \lambda S_t\mathbf{a} \tag{4}$$

To get a stable solution of these generalized eigen-problem, the scatter matrices are required to be non-singular. However, in many real applications such as face recognition, all scatter matrices are usually singular since the data points reside in a very high-dimensional space and in general the sample size is smaller than the dimension. This is known as the *undersampled* problems [27]. There are a lot of extensions proposed in the past try to handle this problem. These approaches include two-stage PCA+LDA [2], Null space LDA [11][24] and LDA/GSVD [23][45][48], *etc*.

C. LPP

Different from PCA and LDA which aim to discover the Euclidean structure, LPP aims to discover the local manifold structure. Given a data affinity (*i.e.*, item-item similarity) matrix W, the optimal projections can be obtained by solving the following minimization problem [20]:

$$\begin{aligned}\mathbf{a}_{opt} &= \arg\min_{\mathbf{a}} \sum_{ij} \left(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right)^2 W_{ij} \\ &= \arg\min_{\mathbf{a}} \mathbf{a}^T XLX^T\mathbf{a} \end{aligned} \tag{5}$$

where $L = D - W$ is the *graph Laplacian* [12] and $D_{ii} = \sum_j W_{ij}$. The matrix D provides a natural measure on the data points. The bigger the value D_{ii} (corresponding to y_i) is, the more "important" is y_i. Therefore, we impose a constraint as follows:

$$\mathbf{y}^T D\mathbf{y} = 1 \Rightarrow \mathbf{a}^T XDX^T\mathbf{a} = 1,$$

where $\mathbf{y} = (y_1, \cdots, y_m)^T = X^T\mathbf{a}$.

Finally, the minimization problem reduces to finding:

$$\mathop{\arg\min_{\mathbf{a}}}_{\mathbf{a}^T XDX^T\mathbf{a}=1} \quad \mathbf{a}^T XLX^T\mathbf{a} \tag{6}$$

The projection vector $\mathbf{a}$ that minimizes the objective function is given by the *minimum* eigenvalue solution to the generalized eigenvalue problem:

$$XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a} \tag{7}$$

It is easy to see that:

$$\begin{aligned} & XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a} \\ \Rightarrow\ & XDX^T\mathbf{a} - XWX^T\mathbf{a} = \lambda XDX^T\mathbf{a} \\ \Rightarrow\ & XWX^T\mathbf{a} = (1-\lambda)XDX^T\mathbf{a} \end{aligned}$$

Therefore, LPPs can also be obtained by solving the following *maximum* eigenvalue problem:

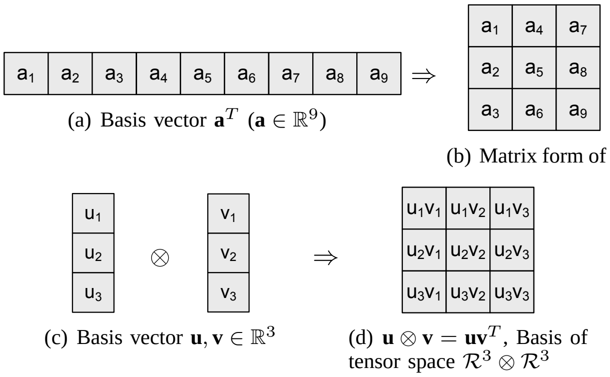
$$XWX^T\mathbf{a} = \lambda XDX^T\mathbf{a} \tag{8}$$

For the detailed derivation of LPP and the choices of W, please see [20].

The choices of different graph structure (different affinity matrix W) play the central role of LPP. Particularly, He *et al.* [21] show that with the following supervised affinity matrix

$$W_{ij} = \begin{cases} 1/m_k, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the } k\text{-th class;} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

LPP will be identical to LDA. For the detailed analysis and experimental comparison of LDA and LPP, please see [21][8].



(a) Basis vector $\mathbf{a}^T$ ($\mathbf{a} \in \mathbb{R}^9$)

(b) Matrix form of **a**

(c) Basis vector $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$

(d) $\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^T$, Basis of tensor space $\mathcal{R}^3 \otimes \mathcal{R}^3$

Fig. 1. Take face images of size 3×3. The ordinary vector-based subspace learning algorithms (*e.g.* PCA, LDA and LPP) first convert the face images to 9-dimensional vectors and compute the basis vectors (projection functions). The basis vector is also 9-dimensional, as shown in (a). (b) The basis vector can be converted to the matrix form and shown as an image, which was referred as Eigenface (PCA), Fisherface (LDA) and Laplacianface (LPP). The 9 numbers in the basis vector are independent estimated and there is no spatial relation between them. (c) The tensor-based subspace learning approaches directly take 3×3 face images as input and compute a set of 3-dimensional basis vectors **u**'s and **v**'s. (d) Each **u** and **v** form a basis $\mathbf{u} \otimes \mathbf{v}$ in tensor space which can also be shown as an image. The 9 numbers in the tensor basis only have 6 degrees of freedom and the values in the same row (column) have a common divisor. However, there is no guarantee of the spatial smoothness of the basis function.

D. Tensor Extensions

A face image represented in the plane is intrinsically a matrix, or the second order tensor. The relationship between nearby pixels of the image might be important for finding a projection. Recently there have been a lot of interest in extending the ordinary vector-based subspace learning approaches (PCA, LDA and LPP) to tensor space [7], [42], [46], [43], [47], [19].

The problem of linear tensor subspace learning is the following. Given a set of data points $T_1, \cdots, T_m$ in $\mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$, find two projection matrices U of size $n_1 \times l_1$ and V of size $n_2 \times l_2$ that maps these m points to a set of points $Y_1, \cdots, Y_m \in \mathcal{R}^{l_1} \otimes \mathcal{R}^{l_2} (l_1 < n_1, l_2 < n_2)$, such that Y_i "represents" T_i, where $Y_i = U^T T_i V$. The difference among tensor extensions of PCA, LDA and LPP lies in their different interpretations on "represents", *i.e.*, different objective functions.

The tensor-based approaches directly operate on the matrix representation of image data and are believed can capture the spatial relationship between the pixels. To examine what kind of spatial relationship has been captured in these tensor-based approaches, we need to examine the basis function.

Let $\{\mathbf{u}_k\}_{k=1}^{n_1}$ be an orthonormal basis of $\mathcal{R}^{n_1}$ and $\{\mathbf{v}_l\}_{l=1}^{n_2}$ be an orthonormal basis of $\mathcal{R}^{n_2}$. It can be shown that $\{\mathbf{u}_i \otimes \mathbf{v}_j\}$ forms a basis of the tensor space $\mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$ [28]. Specifically, the projection of $T \in \mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$ on the basis $\mathbf{u}_i \otimes \mathbf{v}_j$ can be computed as their inner product:

$$< T, \mathbf{u}_i \otimes \mathbf{v}_j >=< T, \mathbf{u}_i\mathbf{v}_j^T >= \mathbf{u}_i^T T\mathbf{v}_j$$

The ordinary vector-based approaches are linear, *i.e.*, $y_i = \mathbf{a}^T\mathbf{x}_i$ where $\mathbf{x}_i \in \mathbb{R}^n$ is the *vector* representation of the i-th image, $\mathbf{a}$ is the projection vector (basis vector) and y_i is the one-dimensional embedding on this basis. The n values in basis function $\mathbf{a}$ are independently estimated. The tensor-based approaches are multilinear, *i.e.*, $y_i = \mathbf{u}^T T_i \mathbf{v} =$ where $T_i \in \mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$ is the *matrix* representation of the i-th image and $n = n_1 \times n_2$. The n values in a tensor basis $\mathbf{u}\mathbf{v}^T$ only have $n_1 + n_2$ degrees of freedom. In fact, the tensor-based approaches can be thought of as special cases of vector-based approaches with the following constraint:

$$a_{i+n_1(j-1)} = u_i v_j \tag{10}$$

where a_i, u_i and v_i are the i-th elements in $\mathbf{a}$, $\mathbf{u}$ and $\mathbf{v}$ respectively.

Figure (1) gives a intuitive example. It is easy to see that there is a common divisor of the values belong to the same row (or column) in a tensor basis, which exactly the spatial relation captured by the tensor-based approaches. Intuitively, the spatial correlation of pixels in a face image would suggest the spatial smoothness of the basis function, *i.e.*, the element values in basis function would be similar if the elements are spatially near. However, the tensor-based approaches have no guarantee on this and the basis function could still be spatially rough.

A more natural measurement of spatial smoothness of basis function could be the sum of the squared difference between nearby elements. In the next section, we will show how to achieve this by incorporating a 2-D discretized laplacian smoothing term in ordinary vector-based approaches.

III. Regularized LPP with Two-Dimensional Discretized Laplacian Smoothing

In this section, we describe how to apply Laplacian penalized functional to measure the smoothness of the basis vectors of the face space, which plays the key role in the regularized LPP with two-dimensional discretized laplacian smoothing algorithm. We begin with a general description of Laplacian smoothing.

A. Laplacian Smoothing

Let f be a function defined on a region of interest, $\Omega \subset \mathbb{R}^d$. The Laplacian operator $\mathcal{L}$ is defined as follows [26]:

$$\mathcal{L}f(\mathbf{t}) = \sum_{j=1}^{d} \frac{\partial^2 f}{\partial t_j^2} \tag{11}$$

The Laplacian penalty functional, denoted by $\mathcal{J}$, is defined by:

$$\mathcal{J}(f) = \int_{\Omega} \left[\mathcal{L}f\right]^2 d\mathbf{t} \tag{12}$$

Intuitively, $\mathcal{J}(f)$ measures the smoothness of the function f over the region Ω. In this paper, our primary interest is in image. An image is intrinsically a two-dimensional signal. Therefore, we take d to be 2 in the following.

B. Discretized Laplacian Smoothing

As we described previously, $n_1 \times n_2$ face images can be represented as vectors in $\mathbb{R}^n, n = n_1 \times n_2$. Let $\mathbf{a}_i \in \mathbb{R}^n$ be the basis vectors (projection functions) obtained by LPP. Without loss of generality, $\mathbf{a}_i$ can also be considered as functions defined on a $n_1 \times n_2$ lattice.

For a face image, the region of interest Ω is a two-dimensional rectangle, which for notational convenience we take to be $[0,1]^2$. A lattice is defined on Ω as follows. Let $\mathbf{h} = (h_1, h_2)$ where $h_1 = 1/n_1$ and $h_2 = 1/n_2$. Ω_h consists of the set of two-dimensional vectors $\mathbf{t}_i = (t_{i_1}, t_{i_2})$ with $t_{i_j} = (i_j - 0.5) \cdot h_j$ for $1 \leq i_j \leq n_j$ and $1 \leq j \leq 2$. There are a total of $n = n_1 \times n_2$ grid points in this lattice. Let D_j be an $n_j \times n_j$ matrix that yields a discrete approximation to $\partial^2/\partial t_j^2$. Thus if $\mathbf{u} = (u(t_1), \cdots, u(t_{n_j}))$ is an n_j-dimensional vector which is a discretized version of a function $u(t)$, then D_j has the property that:

$$[D_j\mathbf{u}]_i \approx \frac{\partial^2 u(t_i)}{\partial t^2}$$

for $i = 1, \cdots, n_j$. There are many possible choices of D_j [6]. In this work, we apply the modified Neuman discretization [33]:

$$D_j = \frac{1}{h_j^2}\begin{pmatrix} -1 & 1 & & & & & 0 \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ 0 & & & & & 1 & -1 \end{pmatrix}$$

Give D_j, a discrete approximation for two-dimensional Laplacian $\mathcal{L}$ is the $n \times n$ matrix:

$$\Delta = D_1 \otimes I_2 + I_1 \otimes D_2 \tag{13}$$

where I_j is $n_j \times n_j$ identity matrix for $j = 1, 2$. $\otimes$ is the kronecker product defined below [22]:

Definition Let A be a $n \times n$ matrix and B be a $m \times m$ matrix. Then the kronecker product of A and B is the $mn \times mn$ block matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nn}B \end{pmatrix}$$

For a $n_1 \times n_2$ dimensional vector $\mathbf{a}$, it is easy to check that $\|\Delta \cdot \mathbf{a}\|^2$ is proportional to the the sum of the squared difference between nearby grid points of $\mathbf{a}$ with its matrix form. It provides a measure of smoothness of $\mathbf{a}$ on the $n_1 \times n_2$ lattice.

C. The Algorithm

Given a pre-defined graph structure with weight matrix W, the Regularized LPP is defined as the minimizer of

$$\sum_{ij} \left(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right)^2 W_{ij} + \alpha\mathcal{J}(\mathbf{a}), \tag{14}$$

where $\mathcal{J}$ is the discretized Laplacian regularization functional:

$$\mathcal{J}(\mathbf{a}) = \|\Delta \cdot \mathbf{a}\|^2 = \mathbf{a}^T \Delta^T \Delta \mathbf{a}. \tag{15}$$

The parameter $\alpha \geq 0$ controls the smoothness of the estimator.

By simple algebraic formulations [20], we have:

$$\sum_{ij} (\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j)^2 W_{ij} = \mathbf{a}^T X L X^T \mathbf{a}.$$

With the same constraint as the standard LPP [20], finally the minimization problem reduces to finding:

$$\underset{\substack{\mathbf{a} \\ \mathbf{a}^T X D X^T \mathbf{a} = 1}}{\arg\min} \quad \mathbf{a}^T \big(X L X^T + \alpha \Delta^T \Delta \big) \mathbf{a}. \tag{16}$$

We will now switch to a Lagrangian formulation of the problem. The Lagrangian is as follows

$$\mathbb{L} = \mathbf{a}^T \left(X L X^T + \alpha \Delta^T \Delta \right) \mathbf{a} - \lambda \mathbf{a}^T X D X^T \mathbf{a}. \tag{17}$$

Requiring that the gradient of $\mathbb{L}$ vanish gives the following eigenvector problem:

$$\left(X L X^T + \alpha \Delta^T \Delta \right) \mathbf{a} = \lambda X D X^T \mathbf{a}. \tag{18}$$

It is easy to show that the matrices XLX^T, XDX^T and $\Delta^T \Delta$ are all symmetric and positive semi-definite. Since $\alpha \geq 0$, the matrix $XLX^T + \alpha \Delta^T \Delta$ is also symmetric and positive semi-definite. The vectors $\mathbf{a}_i$ $(i = 0, 1, \cdots, l-1)$ that minimize the objective function (16) are given by the minimum eigenvalue solutions to the above generalized eigenvalue problem.

D. Model Selection

The $\alpha \geq 0$ is an essential parameter in RLPP model which controls the smoothness of the estimator. When $\alpha = 0$, the RLPP model will reduce to the ordinary LPP which totally ignores the spatial relationship between pixels of an image. When $\alpha \rightarrow \infty$, the RLPP model will choose a spatially smoothest basis vector $\mathbf{a}$ and totally ignore the manifold structure of the face data. RLPP with a suitable α is a trade-off between these two extreme cases. Thus, a natural question would be how to choose the parameter α, or how to select the model.

Model selection is an essential task in most of the learning algorithms [18]. Among various kinds of methods, cross validation is probably the simplest and most widely used one. In this paper, we also use cross validation for model selection. By noticing that the objective function of RLPP can also be written as

$$\underset{\substack{\mathbf{a} \\ \mathbf{a}^T X D X^T \mathbf{a} = 1}}{\arg\min} \quad \mathbf{a}^T \big((1-\beta) X L X^T + \beta \Delta^T \Delta \big) \mathbf{a}. \tag{19}$$

It is easy to see that if we ignore the prior constant, the minimization problems (16) and (19) are equivalent. Now the cross validation can be used for selecting parameter β which is in the interval of $[0, 1]$.

IV. Theoretical Analysis

In this section, we provide some theoretical analysis of the two-dimensional discrete Laplacian Δ as well as our regularized LPP algorithm.

Given a projection vector $\mathbf{a} \in \mathbb{R}^{n_1 \times n_2}$, its spatial smoothness can be measured as $\|\Delta \cdot \mathbf{a}\|$. To remove the impact of the norm of $\mathbf{a}$, we have the following definition:

Definition Let $\mathbf{a} \in \mathbb{R}^n, n = n_1 \times n_2$ be a projection vector. The *Discretized Laplacian Smoothing Function* $\mathcal{S}$ is defined as follows.

$$\mathcal{S}(\mathbf{a}) = \frac{\|\Delta \mathbf{a}\|^2}{\|\mathbf{a}\|^2} = \frac{\mathbf{a}^T \Delta^T \Delta \mathbf{a}}{\mathbf{a}^T \mathbf{a}} \tag{20}$$

$\mathcal{S}(\mathbf{a})$ measures the smoothness of the projection vector $\mathbf{a}$ over the $n_1 \times n_2$ lattice. The smaller $\mathcal{S}(\mathbf{a})$ is, the smoother $\mathbf{a}$ is.

It is easy to see that the "smoothest" $\mathbf{a}$ which minimizes $\mathcal{S}(\mathbf{a})$ is the eigenvector of $\Delta^T \Delta$ corresponding to the smallest eigenvalue. Figure 2 shows the first five and the last five eigenvectors of $\Delta^T \Delta$. The eigenvalues of $\Delta^T \Delta$ are exactly the values of $\mathcal{S}(\mathbf{a})$, where $\mathbf{a}$'s are the corresponding eigenvectors. As can be seen, the first five eigenvectors are spatially smoother than the last five eigenvectors. Particularly, the first eigenvector is a vector of all ones.

We have the following theorem:

Theorem 1: The smallest eigenvalue of $\Delta^T \Delta$ is 0 and the corresponding eigenvector is $\mathbf{e} = (1, \cdots, 1)^T$, which is a vector of all ones.

Proof: $\Delta^T \Delta$ is positive semi-definite. All the eigenvalues of $\Delta^T \Delta$ are non-negative. It is sufficient to show that $\mathbf{e}$ is the eigenvector of $\Delta^T \Delta$ corresponding to eigenvalue 0. We have:

$$\begin{aligned}
\Delta \cdot \mathbf{e} &= (D_1 \otimes I_2 + I_1 \otimes D_2) \cdot \mathbf{e} \\
&= (D_1 \otimes I_2) \cdot \mathbf{e} + (I_1 \otimes D_2) \cdot \mathbf{e} \\
&= \frac{1}{h_1^2} \begin{pmatrix} -I_2 & I_2 & & & 0 \\ I_2 & -2I_2 & I_2 & & \\ & \cdot & \cdot & \cdot & \\ & & I_2 & -2I_2 & I_2 \\ 0 & & & I_2 & -I_2 \end{pmatrix} \mathbf{e} + \\
&\quad \begin{pmatrix} D_2 & & & 0 \\ & D_2 & & \\ & & \cdot & \\ & & & D_2 \\ 0 & & & & D_2 \end{pmatrix} \mathbf{e} \\
&= \mathbf{0} + \mathbf{0} = \mathbf{0}
\end{aligned}$$

Thus,

$$\Delta^T \Delta \cdot \mathbf{e} = \mathbf{0} = 0 \cdot \mathbf{e},$$

$\mathbf{e}$ is the eigenvector of $\Delta^T \Delta$ corresponding to eigenvalue 0. ■

Let λ_{LPP} and λ_{RLPP} be the smallest eigenvalues of equations (7) and (18), respectively,

$$\lambda_{LPP} = \min_{\substack{\mathbf{a} \\ \mathbf{a}^T X D X^T \mathbf{a} = 1}} \mathbf{a}^T X L X^T \mathbf{a} \tag{21}$$

and

$$\lambda_{RLPP} = \min_{\substack{\mathbf{a} \\ \mathbf{a}^T X D X^T \mathbf{a} = 1}} \mathbf{a}^T \left(X L X^T + \alpha \Delta^T \Delta \right) \mathbf{a} \tag{22}$$

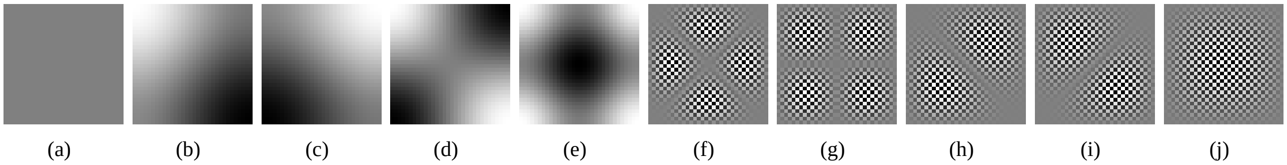


Fig. 2. The first five (a∼e) and last five (f∼j) eigenvectors of $\Delta^T\Delta$. Δ is the discrete approximation for two-dimensional Laplacian as defined in Equation (13). Here $n_1 = n_2 = 32$ and thus Δ is a 1024×1024 matrix. All the eigenvectors are 1024-dimensional vectors and are displayed here as 32×32 images. The smoothness of eigenvectors can be measured by their corresponding eigenvalues. The smaller of the eigenvalue, the smoother of the eigenvector. The first five eigenvectors are spatially smooth while the last five eigenvectors are spatially rough.

Let $\mathbf{a}_{LPP}$ and $\mathbf{a}_{RLPP}$ be the corresponding eigenvectors. By definition (21), it is easy to see that:

$$\mathbf{a}_{LPP}^T XLX^T\mathbf{a}_{LPP} \leq \mathbf{a}_{RLPP}^T XLX^T\mathbf{a}_{RLPP}$$

This indicates that LPP has more locality preserving power than RLPP. As to the smoothness of the eigenvectors, we have the following theorem:

Theorem 2: $\|\Delta \cdot \mathbf{a}_{RLPP}\| \leq \|\Delta \cdot \mathbf{a}_{LPP}\|$

Proof: By definition (21), we have:

$$\mathbf{a}_{LPP}^T XLX^T\mathbf{a}_{LPP} \leq \mathbf{a}_{RLPP}^T XLX^T\mathbf{a}_{RLPP}$$

By definition (22), we have:

$$\begin{aligned} & \mathbf{a}_{RLPP}^T\big(XLX^T + \alpha\Delta^T\Delta\big)\mathbf{a}_{RLPP} \\ \leq\ & \mathbf{a}_{LPP}^T\big(XLX^T + \alpha\Delta^T\Delta\big)\mathbf{a}_{LPP} \\ \leq\ & \mathbf{a}_{RLPP}^T XLX^T\mathbf{a}_{RLPP} + \alpha\mathbf{a}_{LPP}^T\Delta^T\Delta\mathbf{a}_{LPP} \end{aligned}$$

Subtracting $\mathbf{a}_{RLPP}^T XLX^T\mathbf{a}_{RLPP}$ from both sides and noticing that $\alpha > 0$, we get:

$$\begin{aligned} \|\Delta \cdot \mathbf{a}_{RLPP}\|^2 &= \mathbf{a}_{RLPP}^T\Delta^T\Delta\mathbf{a}_{RLPP} \\ &\leq \mathbf{a}_{LPP}^T\Delta^T\Delta\mathbf{a}_{LPP} = \|\Delta \cdot \mathbf{a}_{LPP}\|^2 \end{aligned}$$

■

Theorem (2) indicates that the basis functions obtained by RLPP are spatially smoother than those obtained by LPP.

V. Learning Smooth Laplacianfaces for Representation

Based on Regularized LPP with two-dimensional discretized laplacian smoothing, we describe our *Smooth Laplacianfaces* method for face representation in this section. In recent years, there has been a growing interest in 2-d laplacian smoothing, as well as other higher order smoothing methods. These methods have been used in image de-noising [4], image reconstruction [9], [10] and image warping [16].

In the face analysis and recognition problem, one is confronted with the difficulty that the matrix XDX^T is sometimes singular. This stems from the fact that sometimes the number of images in the training set (m) is much smaller than the number of pixels in each image (n). In such a case, the rank of XDX^T is at most m, while XDX^T is a $n \times n$ matrix, which implies that XDX^T is singular. In the following, we will show how to use Singular Value Decomposition (SVD) to solve this problem.

Suppose $rank(X) = p$, the SVD decomposition of X is

$$X = U\Sigma V^T$$

where $U \in \mathbb{R}^{n\times p}$, $V \in \mathbb{R}^{m\times p}$ and Σ is a $p \times p$ diagonal matrix. Let $\widetilde{X} = U^TX = \Sigma V^T$ and $\mathbf{b} = U^T\mathbf{a}$, we have

$$\mathbf{a}^TXDX^T\mathbf{a} = \mathbf{a}^TU\Sigma V^TDV\Sigma U^T\mathbf{a} = \mathbf{b}^T\widetilde{X}D\widetilde{X}^T\mathbf{b}$$

and

$$\mathbf{a}^TXLX^T\mathbf{a} = \mathbf{a}^TU\Sigma V^TLV\Sigma U^T\mathbf{a} = \mathbf{b}^T\widetilde{X}L\widetilde{X}^T\mathbf{b}.$$

To deal with the smoothing term $\mathbf{a}^T\Delta^T\Delta\mathbf{a}$, we need a small trick. By noticing that given U, $\mathbf{b}$ and equation $U^T\mathbf{a} = \mathbf{b}$, there will be infinitely many solutions of $\mathbf{a}$ which satisfies this equation. Among all these solutions, $\mathbf{a} = U\mathbf{b}$ is obviously one of them and actually is the *minimum norm* solution [34]. If we choose $\mathbf{a} = U\mathbf{b}$, then

$$\mathbf{a}^T\Delta^T\Delta\mathbf{a} = \mathbf{b}^TU^T\Delta^T\Delta U\mathbf{b}$$

Now, the objective function of RLPP can be rewritten as:

$$\underset{\substack{\mathbf{b} \\ \mathbf{b}^T\widetilde{X}D\widetilde{X}^T\mathbf{b}=1}}{\arg\min}\ \mathbf{b}^T\big(\widetilde{X}L\widetilde{X}^T + \alpha U^T\Delta^T\Delta U\big)\mathbf{b}. \tag{23}$$

and the optimal $\mathbf{b}$'s are the minimum eigenvectors of eigen-problem:

$$\Big(\widetilde{X}L\widetilde{X}^T + \alpha U^T\Delta^T\Delta U\Big)\,\mathbf{b} = \lambda\widetilde{X}D\widetilde{X}^T\mathbf{b}.$$

It is easy to check that $\widetilde{X}D\widetilde{X}^T$ is nonsingular and the eigen-problem can be stably solved. After we calculate $\mathbf{b}$, as we mentioned, there are infinitely many solutions of $\mathbf{a}$ which satisfy the equation $U^T\mathbf{a} = \mathbf{b}$. We simply pick the minimum norm solution $\mathbf{a} = U\mathbf{b}$.

It is important to note that if the data are centered, which is a common pre-processing step in most of the learning tasks, the SVD is essentially same as PCA. The left singular vector matrix U is exactly the projection matrix of PCA [17]. Our analysis here also illustrates the rationale behind Laplacianface approach (PCA+LPP).

We formally state the algorithmic procedure of Smooth Laplacianfaces below:

1) **PCA Projection**: We project the face images $\mathbf{x}_i$ into the PCA subspace by throwing away the components corresponding to zero eigenvalue. We denote the projection matrix of PCA by U_{PCA}. We denote as:

$$\widetilde{X} = U_{PCA}^TX \quad \text{and} \quad \widetilde{\Delta} = \Delta \cdot U_{PCA} \tag{24}$$

The PCA projection step can also be viewed as SVD decomposition after centering the data. The rationale of this step is as stated before.

2) **Constructing the Adjacency Graph**: Let G denote a graph with m nodes. The i-th node corresponds to the face image $\mathbf{x}_i$. We put an edge between nodes i and j if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close", i.e. $\mathbf{x}_i$ is among k nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among k nearest neighbors of $\mathbf{x}_i$. Note that, if the class information is available, we simply put an edge between two data points belonging to the same class.
3) **Choosing the Weights**: W is a sparse symmetric $m \times m$ matrix with W_{ij} having the weight of the edge joining vertices i and j. If node i and j are connected, put
$$W_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}$$
Otherwise, put W_{ij} = 0. The weight matrix W of graph G models the local structure of the face manifold. The justification of this weight can be traced back to [3].
4) **Eigenmap**: Compute the eigenvectors and eigenvalues for the generalized eigenvector problem:
$$\left(\widetilde{X}L\widetilde{X}^T + \alpha\widetilde{\Delta}^T\widetilde{\Delta}\right)\mathbf{b} = \lambda\widetilde{X}D\widetilde{X}^T\mathbf{b} \qquad (25)$$
where D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of S, $D_{ii} = \sum_j W_{ji}$. $L = D - W$ is the Laplacian matrix [12].

Let $\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_{l-1}$ be the solutions of (25), ordered according to their eigenvalues, $0 \leq \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{l-1}$. These eigenvalues are equal to or greater than zero because the matrix $\widetilde{X}L\widetilde{X}^T + \alpha\widetilde{\Delta}^T\widetilde{\Delta}$ is symmetric and positive semi-definite and $\widetilde{X}D\widetilde{X}^T$ is symmetric and positive definite. Thus, the embedding is as follows:

$$\mathbf{x} \rightarrow \mathbf{y} = A^T\mathbf{x} \qquad (26)$$

$$A = U_{PCA}B_{RLPP} \qquad (27)$$

$$B_{RLPP} = [\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_{l-1}] \qquad (28)$$

where $\mathbf{y}$ is a l-dimensional vector and A is the projection matrix. This linear mapping not only preserves the manifold's estimated intrinsic geometry in a linear sense but also considers the spatial correlation of image pixels. The column vectors of A are the so-called *Smooth Laplacianfaces*.

VI. Experimental Results

In this section, several experiments are carried out to show the effectiveness of our proposed Smooth Laplacianfaces method for face representation and recognition.

A. Face Representation Using Smooth Laplacianfaces

As we described previously, a face image can be represented as a point in image space. A typical image of size $n_1 \times n_2$ describes a point in $n_1 \times n_2$-dimensional image space. However, due to the unwanted variations resulting from changes in lighting, facial expression, and pose, the image space might not be an optimal space for visual representation.

In Section V, we have discussed how to learn a spatially smooth locality preserving face subspace. The images of faces in the training set are used to learn such a subspace. The subspace is spanned by the column vectors of A in Eqn. (27), i.e., $\mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{l-1}$. We can display the eigenvectors as images. These images may be called *Smooth Laplacianfaces* (S-Laplacianfaces). Using the Yale face database as the training set, we present the first 5 S-Laplacianfaces in Fig. (3), together with Eigenfaces, Fisherfaces and Laplacianfaces. Note that there is a parameter α which controls the smoothness in S-Laplacianfaces. Fig. (3) shows three groups S-Laplacianfaces with $\alpha = 0.5$, 5 and 50. For each face (eigenvector **a**), we also calculated the $\|\Delta \cdot \mathbf{a}\|$. Since each eigenvector is normalized, $\|\Delta \cdot \mathbf{a}\|$ can measure the smoothness of **a** as we discussed in Section (IV). For comparison, we also show the basis of tensor-based approaches in Fig. (3). CSA [42], 2DLDA [47] and TSA [19] are tensor extensions of PCA, LDA and LPP respectively.

We can see that S-Laplacianfaces is smoother than Laplacianfaces. The bigger α is, the smoother are S-Laplacianfaces. The Fisherfaces and Laplacianfaces are somehow similar to each other since they have similar graph structure as we described in Section II. The approaches based on PCA (Eigenfaces and CSA) are the smoothest among all the faces. However, Eigenfaces and CSA do not encode discriminating information thus are not optimal for recognition. As we discussed in Section II-D, the bases of tensor approaches only consider the relationship of pixels in the same row (or column), thus the bases in 2DLDA and TSA are still spatially rough. S-Laplacianfaces consider both the discriminating power and the spatial correlation between the pixels in the face images.

B. Face Recognition Using Smooth Laplacianfaces

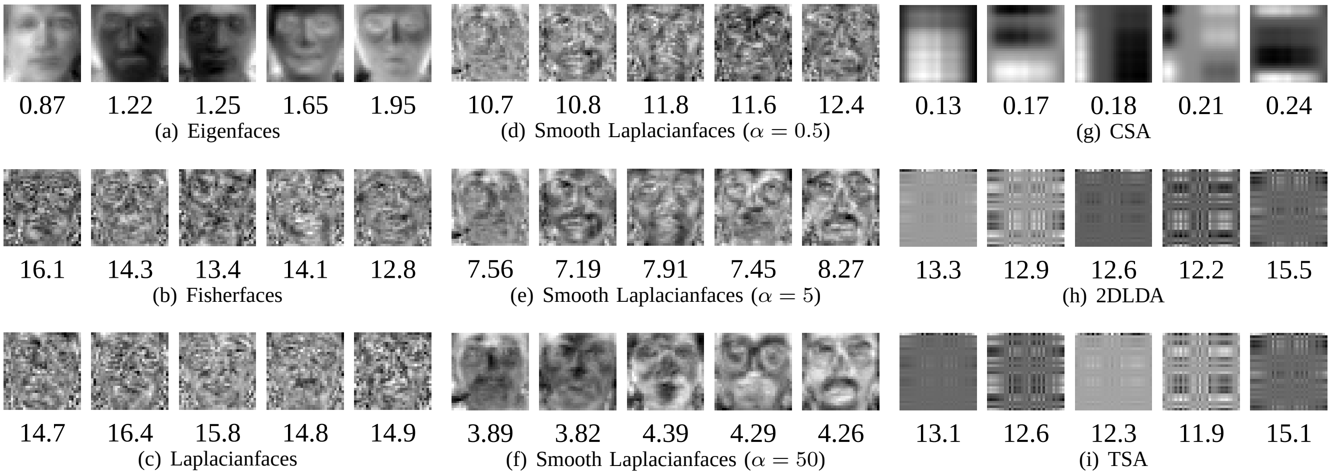
In this section, we investigate the performance of our proposed Smooth Laplacianface method for face recognition.

1) Datasets: Four face databases were used in our experimental study.

- The Yale face database[1] was constructed at the Yale Center for Computational Vision and Control. It contains 165 gray scale images of 15 individuals, each individual has 11 images. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). A random subset with l $(= 2, 3, 4, 5)$ images per individual was taken with labels to form the training set, and the rest of the database was considered to be the testing set.
- The ORL (Olivetti Research Laboratory) face database[2] consists of a total of 400 face images, of a total of 40 people (10 samples per person). The images were captured at different times and have different variations including expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. For each

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[2] http://www.cl.cam.ac.uk/Research/DTG/attarchive/facesataglance.html



(a) Eigenfaces (d) Smooth Laplacianfaces ($\alpha = 0.5$) (g) CSA

(b) Fisherfaces (e) Smooth Laplacianfaces ($\alpha = 5$) (h) 2DLDA

(c) Laplacianfaces (f) Smooth Laplacianfaces ($\alpha = 50$) (i) TSA

Fig. 3. (a) $\sim$ (f) The first 5 Eigenfaces, Fisherfaces, Laplacianfaces, and Smooth Laplacianfaces calculated from the face images in the Yale database. For each face (eigenvector **a**), we also calculated and showed the $\|\Delta \cdot \mathbf{a}\|$ below of each image. Since each eigenvector is normalized, $\|\Delta \cdot \mathbf{a}\|$ can measure the smoothness of **a** as we discussed in Section (IV). S-Laplacianfaces is smoother than Laplacianfaces and Fisherfaces. With bigger α, S-Laplacianfaces become much smoother. (g) $\sim$ (i) The bases of tensor approaches, CSA, 2DLDA and TSA are tensor extensions of PCA, LDA and LPP respectively. The five bases are $\mathbf{u}_1\mathbf{v}_1^T$, $\mathbf{u}_2\mathbf{v}_1^T$, $\mathbf{u}_1\mathbf{v}_2^T$, $\mathbf{u}_2\mathbf{v}_2^T$ and $\mathbf{u}_3\mathbf{v}_1^T$.

individual, $l(= 2, 3, 4, 5)$ images are randomly selected for training and the rest are used for testing.

- The Extended Yale face database B[3] contains 16128 images of 28 human subjects under 9 poses and 64 illumination conditions [15][29]. In this experiment, we choose the frontal pose and use all the images under different illumination, thus we get 64 images for each person. For each individual, $l(= 5, 10, 20, 30)$ images are randomly selected for training and the rest are used for testing.
- The CMU PIE (Pose, Illumination, and Expression) face database[4] contains 68 subjects with 41,368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. We choose the five near frontal poses (C05, C07, C09, C27, C29) and use all the images under different illuminations and expressions, thus we get 170 images for each individual. For each individual, $l(= 5, 10, 20, 30)$ images are randomly selected for training and the rest are used for testing.

In all the experiments, preprocessing to locate the faces was applied. Original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final images for matching. The size of each cropped image in all the experiments is 32×32 pixels, with 256 gray levels per pixel. The features (pixel values) are then scaled to [0,1] (divided by 256). For the vector-based approaches, the image is represented as a 1024-dimensional vector, while for the tensor-based approaches the image is represented as a (32×32)-dimensional matrix, or the second order tensor.

2) Compared algorithms: Many subspace learning algorithms have been proposed and applied for face recognition in the past decades. Among them, PCA, LDA, LPP and there variations are the most popular ones. In our experiments, we compared our Smooth Laplacianface approach with PCA, LDA, LPP and their variations. Particularly, we are interested in their tensor extensions since the motivation of the tensor approaches is also trying to capture the information of spatial relationship between pixels in a face image. The following algorithms are compared in our experiments:

1) Eigenface approach (PCA) [39].
2) Fisherface approach (PCA+LDA) [2].
3) Laplacianface approach (PCA+LPP) [21].
4) Uncorrelated LDA (ULDA), which was recently proposed for extracting feature vectors with uncorrelated attributes [25]. A more recent work [45] showed the equivalence of ULDA and classical LDA, and proposed an efficient algorithm based on Generalized Singular Value Decomposition which can well handle the undersampled problem.
5) Orthogonal LDA (OLDA), which is essentially similar to ULDA except that the discriminant vectors of OLDA are orthogonal to each other [45]. In [48], Ye *et al*. showed the equivalence between OLDA and Null space LDA [11], [24] for the undersampled problem.
6) Concurrent Subspace Analysis (CSA), a tensor extension of PCA [42]. There are also other tensor extensions of PCA, like GPCA [46], TensorPCA [7], *etc*. These algorithm are essentially the same in the sense of objective function.
7) Two-dimensional LDA (2DLDA), a tensor extension of LDA [47]. Some other tensor extensions of LDA include DATER [43], TensorLDA [7].
8) Tensor Subspace Analysis (TSA), the tensor extension of LPP [19].
9) Smooth Laplacianface, our approach proposed in this paper.

Notice that the approaches based on LPP framework (Laplacianface, TensorLPP and Smooth Laplacianface) need to build an affinity graph. We use the same graph in these algorithms which is build based on the label information, *i.e*., put an edge

[3] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html
[4] http://www.ri.cmu.edu/projects/project_418.html

between two nodes if and only if they have the same label.

In short, the recognition process has three steps. First, we calculate the face subspace from the training set of face images; then the new face image to be identified is projected into d-dimensional subspace (vector-based approaches) or $(d \times d)$-dimensional tensor subspace (tensor-based approaches); finally, the new face image is identified by nearest neighbor classifier.

3) Face recognition results: The recognition error rates of different algorithms on Yale, ORL, Yale-B and PIE datasets are reported on the Table (I), (II), (III) and (IV) respectively. For each given l (the number of training images per individual), we average the results over 50 random splits and report the mean as well as the standard deviation. The cross validation in the training set was used to select the parameter α in our S-Laplacianface algorithm.

A crucial problems for most of the subspace learning based face recognition methods is dimensionality estimation. The performance usually varies with the number of dimensions. We show the best results obtained by different algorithms and the dimension of the corresponding face subspaces are called optimal face subspace thereafter. Note that for LDA and its variations (except TensorLDA), the upper bound of the dimensionality is $c-1$ where c is the number of individuals [2]. For the baseline method, the recognition is simply performed in the original 1024-dimensional image space without any dimensionality reduction.

The main observations from the performance comparisons include:

- S-Laplacianface outperforms the other methods with different numbers of training samples per individual in all the four databases. The reason lies S-Laplacianface explicitly takes into account the spatial relationship between the pixels in an image. The use of spatial information significantly reduces the number of degrees of freedom. Therefore, S-Lapalcianface can have good performance even when there is only a small number of training samples available.
- The methods based on PCA (Eigenface and CSA) perform the worst in most the cases. They fail to gain much improvement over the baseline method. This is probably due to the fact that the PCA is unsupervised and does not encode discriminating information.
- The performances of three variations of LDA, Fisherface, ULDA and OLDA, are different. ULDA and OLDA outperform Fisherface in most cases, especially when there are only a few training examples, *e.g.* 2 train in Yale and ORL databases. To guarantee the non-singularity of the within-class scatter matrix, Fisherface approach first projects the face images to $m-c$ dimensional subspace by using PCA, where m is the number of training images and c is the number of classes. When m is small (only a few training examples), the PCA step which only keeps $m-c$ principle components tends to lose a lot of information. ULDA and OLDA directly solve the objective function of LDA in the original space thus avoid such problem [45]. Our experimental results are consistent with previous studies on extensions of LDA [11], [24], [45]. The OLDA is slightly better than ULDA which may be related to the effect of the noise removal inherent of OLDA as pointed out in [45].
- Two groups of algorithms (ULDA vs. Laplacianface, 2DLDA vs. TSA) perform comparatively to each other, which is not surprising since LPP reduces to LDA with an affinity matrix as in Eq. (9). In our experiments, the only difference between the affinity matrix used in Laplacianface (or TSA) with the matrix in Eq. (9) is the weight on each edge.
- The tensor-based algorithms (2DLDA and TSA) show their advantages in three databases (ORL, Yale-B and PIE) while failed gain improvement on Yale database. This suggests that the spatial relationship of face images considered in tensor-based approach (relation between the pixels in the same row or column) has its limitation. Compare to the tensor approaches, our Smooth Laplacianface is a more natural extension of incorporating spatial information in vector-based algorithm, which is supported by the experimental results.
- It is interesting to note that S-Laplacianface reaches the best performance almost always at $c-1$ dimensions. This property shows that S-Laplacianface does not suffer from the problem of dimensionality estimation. Moreover, such property makes efficient cross validation for model selection in S-Laplacianface possible. In cross validation stage, we simple evaluate the performance on dimension $c-1$ and choose the best parameter α.

4) Model selection for Smooth Laplacianface: The $\alpha \geq 0$ is an essential parameter in our Smooth Laplacianface algorithm which controls the smoothness of the estimator. We use cross validation on the training set to select this parameter in the previous experiments. In this subsection, we try to examine the impact of parameter α on the recognition performance of S-Laplacianface.

Figure (4), (5), (6) and (7) show the performance changing of S-Laplacianface with the parameter α on Yale, ORL, Yale-B and PIE respectively. For convenience, the X-axis is plotted as $\alpha/(1+\alpha)$ which is strictly in the interval $[0,1]$. Each figure has three lines. The curve shows the test error of S-Laplacianface with respect to $\alpha/(1+\alpha)$. The solid line shows the test error of S-Laplacianface with $\alpha = 0$ which is exactly the ordinary Laplacianface algorithm. The dashed line shows the performance of S-Laplacianface with α selection by cross validation on training set.

It is easy to see that S-Laplacianface can achieve significantly better performance than Laplacianface over a large range of α. Thus, the parameter selection is not a very crucial problem in Smooth Laplacinface algorithm. The cross validation for parameter selection achieves a reasonable good result, especially when the training set is large.

VII. Conclusions

In this paper, we propose a new linear dimensionality reduction method called Regularized Locality Preserving Projections (RLPP). RLPP explicitly considers the spatial relationship between the pixels in images. By introducing a

TABLE I
RECOGNITION ERROR RATES ON YALE DATABASE (MEAN±STD-DEV%)

Method	2 Train		3 Train		4 Train		5 Train	
	error	dim	error	dim	error	dim	error	dim
Baseline	54.0±3.3	1024	48.2±3.7	1024	45.1±3.8	1024	41.9±4.0	1024
Eigenfaces	54.0±3.3	29	48.2±3.7	44	45.1±3.8	158	41.9±4.0	74
CSA	50.5±3.4	6^2	45.0±3.4	6^2	42.7±3.8	5^2	38.7±4.3	5^2
Fisherfaces	56.2±3.7	9	39.3±4.1	14	31.4±4.7	14	25.9±4.3	14
ULDA	44.5±3.9	14	33.5±4.1	14	27.4±4.7	14	23.9±3.1	14
OLDA	44.7±4.3	14	33.0±3.7	14	27.2±4.3	14	22.8±3.8	14
2DLDA	54.6±6.9	6^2	42.2±4.7	6^2	37.1±5.6	7^2	33.2±4.3	5^2
Laplacianfaces	44.6±4.1	14	33.6±3.8	14	27.2±4.6	19	23.1±3.6	23
TSA	53.8±6.9	7^2	42.2±4.7	6^2	37.4±5.4	5^2	33.8±4.2	5^2
S-Laplacianfaces*	**43.5±4.3**	18	**32.0±3.6**	14	**25.5±4.6**	14	**21.2±3.2**	14
	43.6±4.3	14	**32.0±3.6**	14	**25.5±4.6**	14	**21.2±3.2**	14

* The first row of S-Laplacianfaces indicates the best performance as well as the optimal subspace dimension. The second row indicates the performance of S-Laplacianfaces at exactly $c - 1$ dimension, c is the number of class.

TABLE II
RECOGNITION ERROR RATES ON ORL DATABASE (MEAN±STD-DEV%)

Method	2 Train		3 Train		4 Train		5 Train	
	error	dim	error	dim	error	dim	error	dim
Baseline	29.6±3.1	1024	21.1±2.5	1024	15.5±2.1	1024	11.9±2.1	1024
Eigenfaces	29.6±3.1	79	21.1±2.5	119	15.5±2.1	158	11.9±2.1	189
CSA	28.8±3.1	17^2	20.6±2.4	16^2	15.1±1.9	5^2	11.5±2.3	16^2
Fisherfaces	24.5±3.3	28	13.7±2.4	39	8.8±2.0	39	6.1±1.5	39
ULDA	20.0±2.7	39	12.5±2.1	39	8.2±1.8	39	6.0±1.5	39
OLDA	18.2±3.1	39	10.2±2.1	39	6.6±1.5	39	4.5±1.3	39
2DLDA	19.6±3.3	9^2	10.5±2.2	10^2	6.9±1.9	9^2	4.7±1.7	10^2
Laplacianfaces	20.1±2.7	39	12.8±2.2	39	8.7±1.6	39	6.3±1.7	39
TSA	19.6±3.3	9^2	10.5±2.2	10^2	6.8±1.9	9^2	4.8±1.7	10^2
S-Laplacianfaces*	**15.6±2.8**	77	**8.7±1.7**	113	**4.9±1.5**	82	**3.0±1.2**	39
	16.6±2.8	39	**9.0±1.8**	39	**5.0±1.5**	39	**3.0±1.2**	39

* The first row of S-Laplacianfaces indicates the best performance as well as the optimal subspace dimension. The second row indicates the performance of S-Laplacianfaces at exactly $c - 1$ dimension, c is the number of class.

Laplacian penalized functional, the projection vectors obtained by RLPP can be smoother than those obtained by the ordinary LPP. This prior information significantly reduces the number of degrees of freedom, and hence RLPP can perform better than LPP. We applied our RLPP method to face recognition and compared with Eigenface, Fisherface, Laplacianface and their tensor extensions methods on Yale, ORL, PIE, and Yale-B databases. Experimental results show that our method consistently outperforms the other methods.

The primary focus of this paper is on images which are two-dimensional signals. However, the analysis and algorithm presented here can also be naturally extended to higher dimensional signals. For example, a video can be considered as a three-dimensional signal, and thus a three-dimensional discretized Laplacian functional can be applied to video. Other examples include Bidirectional Texture Function (BTF) which is a six-dimensional signal. We are currently investigating the applicability of our algorithm for these problems.

REFERENCES

[1] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[2] Peter N. Belhumeur, J. P. Hepanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, Cambridge, MA, 2001.

[4] Mark Berman. Automated smoothing of image and other regularly spaced data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):460–468, 1994.

[5] M. Brand. Charting a manifold. In *Advances in Neural Information Processing Systems 16*, 2003.

[6] B. L. Buzbee, G. H. Golub, and C. W. Nielson. On direct methods for solving poisson's equations. *SIAM Journal on Numerical Analysis*, 7(4):627–656, Dec. 1970.

[7] Deng Cai, Xiaofei He, and Jiawei Han. Subspace learning based on tensor analysis. Technical report, Computer Science Department, UIUC, UIUCDCS-R-2005-2572, May 2005.

[8] Deng Cai, Xiaofei He, and Jiawei Han. Using graph model for face analysis. Technical report, Computer Science Department, UIUC, UIUCDCS-R-2005-2636, September 2005.

[9] Antonin Chambolle and Pierre-Louis Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76:167–188, 1997.

[10] Tony Chan, Antonio Marquina, and Pep Mulet. High-order total variation-based image restoration. *SIAM J. Sci. Comput.*, 22(2):503–516, 2000.

TABLE III
Recognition error rates on extended Yale database B (mean±std-dev%)

Method	5 Train		10 Train		20 Train		30 Train	
	error	dim	error	dim	error	dim	error	dim
Baseline	69.2±1.4	1024	55.5±1.0	1024	42.1±1.2	1024	34.6±1.2	1024
Eigenfaces	69.2±1.4	189	55.5±1.0	378	42.1±1.2	616	34.6±1.2	780
CSA	69.2±1.4	32^2	55.5±1.0	32^2	42.1±1.2	32^2	34.6±1.2	32^2
Fisherfaces	34.9±2.3	37	21.7±1.2	37	14.1±0.8	37	18.7±1.6	37
ULDA	33.2±4.7	37	23.1±2.9	37	25.9±7.8	37	17.8±1.6	37
OLDA	31.9±2.7	37	20.9±1.5	37	16.4±1.8	37	13.7±1.1	37
2DLDA	33.3±2.7	10^2	21.6±1.6	11^2	14.1±1.2	12^2	10.6±0.9	13^2
Laplacianfaces	32.1±3.9	71	19.5±1.9	76	18.0±4.0	75	13.6±1.2	76
TSA	30.2±1.8	14^2	17.7±1.3	16^2	10.7±0.9	15^2	7.9±0.7	15^2
S-Laplacianfaces*	**29.3±2.0**	69	**16.2±1.3**	280	**8.2±0.9**	273	**5.0±0.7**	387
	29.6±2.1	37	**16.5±1.2**	37	**8.6±0.9**	37	**5.5±0.7**	37

* The first row of S-Laplacianfaces indicates the best performance as well as the optimal subspace dimension. The second row indicates the performance of S-Laplacianfaces at exactly $c-1$ dimension, c is the number of class.

TABLE IV
Recognition error rates on PIE database (mean±std-dev%)

Method	5 Train		10 Train		20 Train		30 Train	
	error	dim	error	dim	error	dim	error	dim
Baseline	76.6±0.6	1024	64.8±0.7	1024	48.6±0.7	1024	37.9±0.6	1024
Eigenfaces	76.6±0.6	334	64.8±0.7	654	48.6±0.7	982	37.9±0.6	1023
CSA	76.6±0.6	32^2	64.8±0.7	32^2	48.6±0.7	32^2	37.9±0.6	32^2
Fisherfaces	42.8±1.7	67	29.7±1.3	67	21.5±0.8	67	10.9±0.4	67
ULDA	37.9±1.5	67	31.8±1.1	67	20.5±0.8	67	10.9±0.5	67
OLDA	38.2±1.6	67	29.1±1.2	67	18.6±0.8	67	10.8±0.6	67
2DLDA	37.4±1.6	12^2	24.6±1.1	11^2	14.8±0.8	13^2	10.5±0.6	13^2
Laplacianfaces	38.0±1.5	67	29.6±1.0	139	20.2±0.8	146	10.8±0.4	86
TSA	38.8±1.6	11^2	26.0±1.1	11^2	15.6±0.7	13^2	11.0±0.5	14^2
S-Laplacianfaces*	**33.0±1.8**	294	**18.1±1.1**	280	**10.4±0.6**	76	**8.2±0.6**	77
	33.4±1.8	67	**18.5±1.1**	67	**10.5±0.6**	67	**8.3±0.6**	67

* The first row of S-Laplacianfaces indicates the best performance as well as the optimal subspace dimension. The second row indicates the performance of S-Laplacianfaces at exactly $c-1$ dimension, c is the number of class.

[11] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000.

[12] Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2000.

[14] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, 1990.

[15] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[16] C. A. Glasbey and K. V. Mardia. A review of image-warping methods. *Journal of Applied Statistics*, 25(2):155–171, 1997.

[17] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[19] Xiaofei He, Deng Cai, and Partha Niyogi. Tensor subspace analysis. In *Advances in Neural Information Processing Systems 18*, 2005.

[20] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2003.

[21] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[22] Roger Horn and Charles Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[23] Peg Howland and Haesun Park. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):995–1006, 2004.

[24] Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma. Solving the small sample size problem of LDA. In *16th International Conference on Pattern Recognition (ICPR'02)*, 2002.

[25] Z. Jin, J. Y. Yang, Z. S. Hu, and Z. Lou. Face recognition based on the uncorrelated discriminant transformation. *Pattern Recognition*, 34:1405–1416, 2001.

[26] Jurgen Jost. *Riemannian Geometry and Geometric Analysis*. Springer-Verlag, 2002.

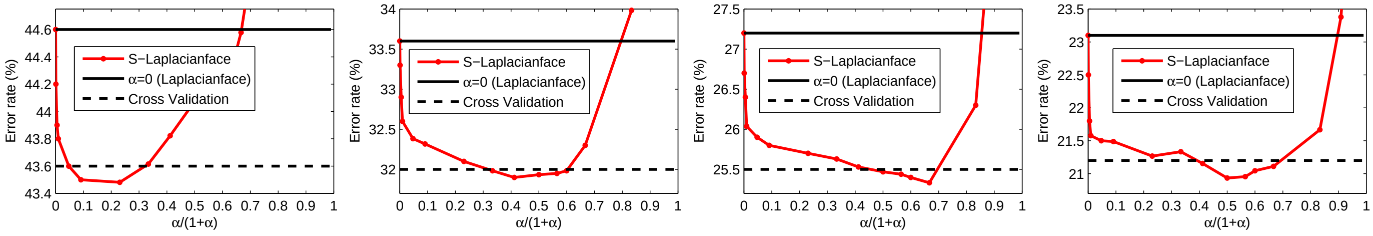
[27] W. J. Krzanowski, P. Jonathan, W. V. McCarthy, and M. R. Thomas. Discriminant analysis with singular covariance matrices: Methods and applications to spectroscopic data. *Applied Statistics*, 44(1):101–115, 1995.

[28] John M. Lee. *Introduction to Smooth Manifolds*. Springer-Verlag New York, 2002.

[29] K.C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

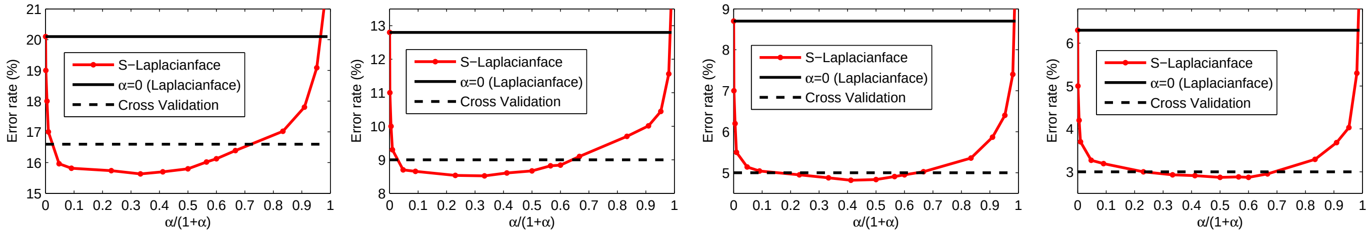
[30] Z. Li, W. Liu, D. Lin, and X. Tang. Nonparametric subspace analysis for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2005.

[31] X. Liu, Y. Yu, and H.-Y. Shum. Synthesizing bidirectional texture functions for real-world surfaces. Number 20, pages 97–106, Los Angeles, CA, 2001.



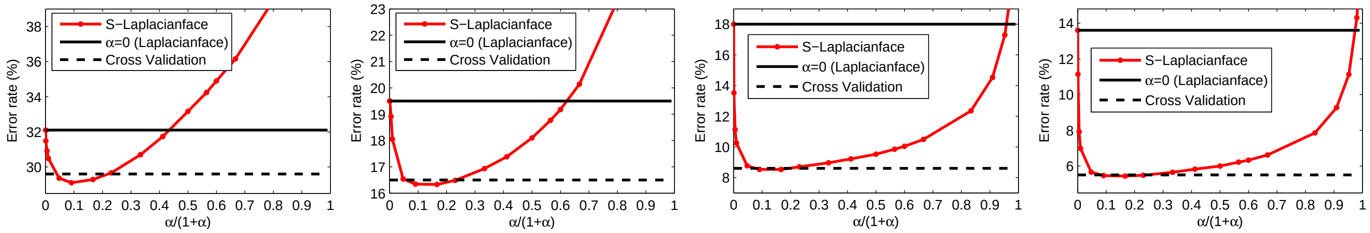
(a) 2 Train (b) 3 Train (c) 4 Train (d) 5 Train

Fig. 4. Model selection on Yale database. The curve shows the test error of S-Laplacianface with respect to $\alpha/(1+\alpha)$. The solid line shows the test error of S-Laplacianface with $\alpha = 0$ which is exactly the ordinary Laplacianface approach. The dashed line shows the performance of S-Laplacianface with α selection by cross validation on training set.



(a) 2 Train (b) 3 Train (c) 4 Train (d) 5 Train

Fig. 5. Model selection on ORL database. The curve shows the test error of S-Laplacianface with respect to $\alpha/(1+\alpha)$. The solid line shows the test error of S-Laplacianface with $\alpha = 0$ which is exactly the ordinary Laplacianface approach. The dashed line shows the performance of S-Laplacianface with α selection by cross validation on training set.



(a) 5 Train (b) 10 Train (c) 20 Train (d) 30 Train

Fig. 6. Model selection on extended Yale database B. The curve shows the test error of S-Laplacianface with respect to $\alpha/(1+\alpha)$. The solid line shows the test error of S-Laplacianface with $\alpha = 0$ which is exactly the ordinary Laplacianface approach. The dashed line shows the performance of S-Laplacianface with α selection by cross validation on training set.

[32] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[33] Finbarr O'Sullivan. Discretized laplacian smoothing by fourier methods. *Journal of the American Statistical Association*, 86(415):634–642, Sep. 1991.

[34] Roger Penrose. On best approximate solution of linear matrix equations. In *Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19, 1956.

[35] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[36] B. Scholkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[37] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[38] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[39] M. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii, 1991.

[40] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis for image ensembles. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[41] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1222–1228, September 2004.

[42] Dong Xu, Shuicheng Yan, Lei Zhang, Hong-Jiang Zhang, Zhengkai Liu, and Heung-Yeung Shum. Concurrent subspace analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[43] Shuicheng Yan, Dong Xu, Qiang Yang, Lei Zhang, Xiaoou Tang, and Hong-Jiang Zhang. Discriminant analysis with tensor representation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'05)*, pages 526–532, 2005.

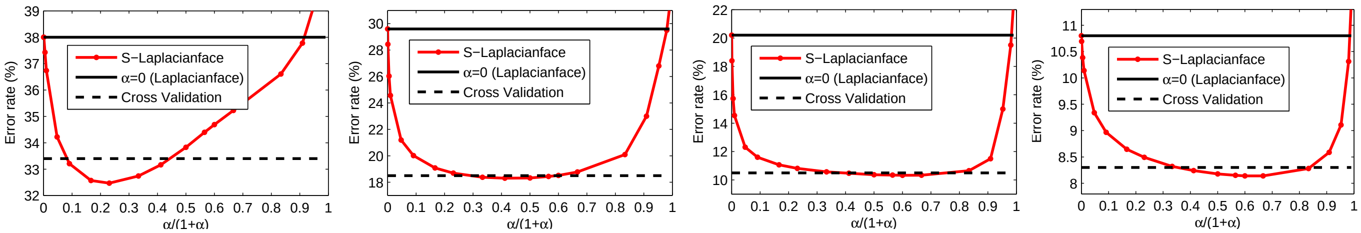
[44] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proc. of the fifth International Conference on Automatic Face and Gesture Recognition*, Washington, D. C., May 2002.

[45] Jieping Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, (6):483–502, 2005.

[46] Jieping Ye, Ravi Janardan, and Qi Li. GPCA: An efficient dimension reduction scheme for image compression and retrieval. In *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[47] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems 17*, 2004.

[48] Jieping Ye and Tao Xiong. Null space versus orthogonal linear



(a) 5 Train

(b) 10 Train

(c) 20 Train

(d) 30 Train

Fig. 7. Model selection on PIE database. The curve shows the test error of S-Laplacianface with respect to $\alpha/(1+\alpha)$. The solid line shows the test error of S-Laplacianface with $\alpha = 0$ which is exactly the ordinary Laplacianface approach. The dashed line shows the performance of S-Laplacianface with α selection by cross validation on training set.

discriminant analysis. In *The Twenty-Third International Conference on Machine Learning*, 2006.

[49] Hongyuan Zha and Zhenyue Zhang. Isometric embedding and continuum isomap. In *Proc. of the twentieth Internation Conference on Machine Learning*, 2003.